

# Client-side Web Mining for Community Formation in Peer-to-Peer Environments

Kun Liu, Kanishka Bhaduri, Kamalika Das, Phuong Nguyen and Hillol Kargupta

Department of Computer Science and Electrical Engineering

University of Maryland Baltimore County

1000 Hilltop Circle, Baltimore, MD 21250

✉ kunliu1, kanishk1, kdas1, phuong3, hillol@cs.umbc.edu

## ABSTRACT

In this paper we present a framework for forming interests-based Peer-to-Peer communities using client-side web browsing history. At the heart of this framework is the use of an order statistics-based approach to build communities with hierarchical structure. We have also carefully considered privacy concerns of the peers and adopted cryptographic protocols to measure similarity between them without disclosing their personal profiles. We evaluated our framework on a distributed data mining platform we have developed. The experimental results show that our framework could effectively build interests-based communities.

## Keywords

Peer-to-Peer community, Order Statistics, Privacy Preserving Data Mining

## 1. INTRODUCTION

According to Maslow's theory [17], social motive, which drives people to seek contact with others and to build satisfying relations with them, is one of the most basic needs of human beings. The tendency to have affiliations with others is visible even in virtual environments such as the World Wide Web. Many online communities like Google and Yahoo groups provide the user a place to share knowledge, and to request and offer services. These communities are usually implemented as forums or mailing lists and under certain central control. As the Web continues to grow in both contents and the number of connected devices, Peer-to-Peer (P2P) distributed computing is becoming increasingly popular. Applications like Napster, KaZaA, BitTorrent, and SETI have already demonstrated the power of such computation. Peer-to-Peer technologies harness the CPUs and storage devices over the network to produce huge data stores, processing engines and communications sys-

Hillol Kargupta is also affiliated with AGNIK, LLC, USA.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists requires prior specific permission and/or a fee.

WEBKDD'06, August 20, 2006, Philadelphia, Pennsylvania, USA.  
Copyright 2006 ACM 1-59593-444-8/06 \$5.00.

tems. Each peer in the P2P environment acts as an autonomous and independent agent that shares knowledge by submitting queries and by replying with relevant information. Dynamically aggregating peers with similar interests could greatly enhance the capability of each individual, facilitate knowledge sharing, and reduce the network load. For example, a peer community allows the establishment of an abstract region of specialization. When a peer needs some relevant resources, the query could be propagated to the community members first to avoid the flooding of the request, and to maximize the quality of search results.

In this paper we address the problem of forming interests-based communities in a Peer-to-Peer environment. We define a Peer-to-Peer community as a collection of nodes in the network that share common interests. Traditional web mining has spent lots of efforts on the web server side, e.g. to analyze the server log. Instead, in this paper, we propose the usage of client-side information, namely, the web browsing cache, to model a peer's personal interests and to build Peer-to-Peer communities. Compared with other related work, our framework has the following specific features:

It makes use of an order statistics-based approach to build communities with hierarchical structures.

It considers privacy concerns of each peer, and adopts cryptographic protocols to measure similarity between peers without disclosing their personal profiles.

Any technique that creates and represents a peer's personal profile as a feature vector can be plugged into our framework.

The remainder of this paper is organized as follows. Section 2 offers an overview of the literature on Peer-to-Peer community formation, Peer-to-Peer data mining, and privacy issues in Peer-to-Peer network. Section 3 presents some basic features of our Peer-to-Peer community framework. Section 4 and 5 address the community formation process. Section 6 discusses the message complexity of some key steps of the formation process. Section 7 studies the performance of the proposed framework and provides the experimental results. Finally, Section 8 concludes this paper with several directions for future work.

## 2. RELATED WORK

This section presents a brief overview of the literature on the formation of Peer-to-Peer communities, Peer-to-Peer

data mining, and privacy in Peer-to-Peer network. Due to the large volume of the literature we do not attempt a comprehensive citation listing. Instead we provide a sampling from a group of major categories.

## 2.1 Peer-to-Peer Communities

Generally speaking, the research on self-formation of Peer-to-Peer communities can be grouped into four major categories: 1) ontology matching-based approach; 2) attribute similarity-based approach; 3) trust-based approach; and 4) link analysis-based approach. We introduce each of them as follows.

Castano and Montanelli addressed the problem of formation of semantic Peer-to-Peer communities [4]. Each peer is associated with an ontology which gives a semantically rich representation of the interests that the peer exposes to the network, in terms of concepts, properties and semantic relations. Each peer interacts with others by submitting discovery queries in order to identify the potential members of an interest-based community, and by replying to incoming queries whether it can join a community. A semantic matchmaker is employed to check whether two peers share the same interests. The matchmaker performs dynamic ontology matching by taking into account both linguistic and contextual features of the concepts to be compared. The advantage of this approach is that peers do not have to agree on the same predefined ontology, and therefore they have lots of flexibility of describing their interests. However, the gain of flexibility comes at the price of accuracy because of the uncertainty of concepts. We refer the reader to [19] for a brief survey of existing ontology matching approaches. The other drawback of this approach is that a peer's interests are inevitably revealed, even to the peers that do not belong to the community, therefore the privacy of the peer is compromised.

Khambatti et al. proposed a Peer-to-Peer community discovery approach where each peer is associated with a set of attributes that represent the interests of that peer [15]. These attributes are chosen from a controlled vocabulary that each peer agrees with, which gets rid of the uncertainty of the fuzzy ontology matching. Peers whose attributes have non-empty intersection can be grouped together. A very basic privacy policy is applied such that a peer does not disclose attributes corresponding to its private interests. This means that the smaller the number of claimed attributes, the smaller the number of communities or community members discovered by a peer. In this paper, we also assume each peer has a set of attributes, which we call as profile vector. The difference is that each interest in the profile vector can be given a weight to show its importance. Moreover, we do not simply check the intersection of attributes, instead, we quantitatively compute the similarity between profile vectors (using scalar product), and we use an order statistics-based algorithm that can tell how similar a pair of peers are to each other in the whole network. Our privacy management scheme enables each peer to measure the similarity with other peer without worrying about privacy breach.

Trust-based community formation is usually discussed in the scenario of file sharing and service providing. The notation "trust" is a measure used by a peer to evaluate other peer's capability of providing a good quality service or resource. This trust is based on information about the peer's

past behavior. Once a peer finds trustworthy peers, it invites them to join its community. We refer the reader to [27, 1] as a starting point on this topic. In this paper, we are interested in forming a community based on peers' interests without considering the past interactions of peers.

There exists another area of research that focuses on the link structure analysis of network to identify patterns of interaction. For example, Scott identified the various cliques, components and circles into which networks are formed [22]. Flake et al. described an approach to identify web communities [8]. Here a web community is a collection of web pages in which each member page has more hyperlinks within the community than outside it. Such communities help to create improved search engines, to perform content filtering, etc. The drawback of link analysis-based approach is that it depends on the stable link structure of the network, and therefore precludes a peer from being a member of more than one community simultaneously.

## 2.2 Peer-to-Peer Data Mining

Peer-to-Peer data mining is a relatively new field. It pays careful attention to the distributed resources of data, computing, communication, and human factors in order to use them in a near optimal fashion. Wolf et al. proposed algorithms for association rule mining [29] and local L2 norm monitoring [28] over P2P networks. Datta et al. proposed an algorithm for K-Means clustering over large, dynamic networks [5].

## 2.3 Privacy in Peer-to-Peer Network

The objective of large scale distributed network is to maximize the availability and utilization of information. This goal would be achieved if the free flow of information was ensured, and if the owners of different data resources were able to share the data with each other. However, this is frequently restricted by legal obligations or by commercial and personal privacy concerns. Privacy, or lack of it, is becoming an increasingly important issue in many distributed application scenarios including file sharing, cooperative computation, etc. Previous research on privacy in Peer-to-Peer network can be roughly classified into two categories: 1) user anonymity; and 2) data privacy.

User anonymity aims at offering the users privacy protection by letting them hide their identities from the communicating peers or from malicious eavesdroppers. There are many uses of anonymous P2P technology that help internet users surf the web anonymously and shield their online activities from corporate or government eyes. Anonymous communication system is also used by government for intelligence gathering and politically sensitive negotiations. Usually a special protocol for anonymous routing is applied in the network (see e.g. [2]). The anonymity comes from the idea that no one knows who requested the information as it is difficult (if not impossible) to determine whether a user requested the data for himself or simply requested the data on behalf of somebody else. The end result is that everybody on the network acts as a universal sender and universal receiver to maintain anonymity. There are many decentralized anonymous and censorship-resistant P2P frameworks in the market such as the Freenet [9] and the GNUnet [11], to name a few.

The objective of protecting data privacy is to hide the sensitive information owned by a peer from being disclosed

in a cooperative computation environment, where the revelation of a peer's identity is unavoidable. For example, it may not be possible to hide the identity ( e.g. IP, port number, URI) of a peer in a Peer-to-Peer community since without this information, peers may not be able to communicate with each other. To be more specific, the data privacy problem in a large scale cooperative computation environment can be defined as follows. Assume that  $n$  participants  $P = \{P_1; P_2; \dots; P_n\}$ , each owning a private input  $x_i$ , wish to jointly compute the output  $f(x_1; x_2; \dots; x_n)$  of some common function  $f$ , without revealing anything but the output. Privacy preserving data mining (PPDM) [26] strives to provide a solution to this problem. It aims to allow useful data patterns to be extracted without compromising privacy. For example, Gilburd et al. presented a privacy model called  $k$ -TTP for large-scale distributed environment [10]. The intuition is that at any time each participant can only learn a combined statistics of a group of at least  $k$  participants, and therefore any specific participant's private input is hidden among at least  $k - 1$  other participants' input. In Section 5.3 we will revisit this problem and discuss how to compute the scalar product of two private vectors owned by two peers.

### 3. FEATURES OF PEER-TO-PEER COMMUNITY

In this section, we present some features that characterize the formation of our Peer-to-Peer communities.

#### 3.1 Peer Profiles

A crucial issue in forming Peer-to-Peer communities is to create peer profiles that accurately reflect a peer's interests. These interests can be either explicitly claimed by a peer, or implicitly discovered from the peer's behaviors. A peer's profile is usually represented by a keyword/concept vector. Trajkova and Gauch proposed techniques to implicitly build ontology-based user profiles by automatically monitoring the user's browsing habits [25]. The system classifies each web page the user has visited into the most similar concept in a predefined hierarchy of ontology. Each element of the user profile vector corresponds to the weight or the number of pages associated with that concept in the ontology. The Open Directory Project concept hierarchy<sup>1</sup> was used as the reference ontology. Figure 1 shows a sample ontology for user profile. Other sources of information have also been used in the literature to create profiles, such as using bookmarks [24], using queries and search results [23], etc. We refer the reader to [25] for a brief overview on this topic.

We point out that any approach that represents a peer's profile in a feature vector can be used in our framework. In this paper, we use the frequencies of the web domains a peer has visited during a period of time to construct the peer's profile vector. Each web domain can be viewed as an interest or topic and hence the frequency represents the weight of the interest for that topic. Detailed explanation about data collection is given in Section 7. To avoid the uncertainty of ontology matching, we expect all peers to agree on the same ontology defined by a controlled vocabulary. In this paper, this means that all peers agree on a superset of web domain names.

#### 3.2 Similarity Measurement

<sup>1</sup>Open Directory Project { <http://dmoz.org/>

Figure 1: A sample ontology for user profile.

The goal of community formation is to find peers sharing similar interests. However, one of the important questions is how a peer can decide whether another peer is similar to him, to what extent? If we simply choose a similarity measurement and setup a subjective threshold such that peers with similarities greater than this threshold can be grouped together, we can't provide any statistical guarantee. Furthermore, this approach is not able to reflect the essential characteristics of a social community, namely, hierarchy. In a social network, a person may have multi-level friends, where the first level might be family members and closest friends, the second level might be some other colleagues. A person could also have indirect friends from his/her friend's social network. A Peer-to-Peer community from one peer's perspective should also have such kind of hierarchical structure. That is, some peers share more interests with this peer, and some less, under some similarity measurement.

To achieve this goal, we propose an order statistics-based approach (to be described later in Section 5.1) that enables a peer to know how similar the other peer is to himself. In other words, our statistical measurement guarantees that if the similarity between peer  $P_i$  and  $P_j$  is above a threshold,  $P_i$  can determine with confidence level  $q$  that  $P_j$  is among the top  $(1 - p)$  quantile most similar peers of  $P_i$ 's. Here the quantile, denoted by  $p$  with  $0 < p < 1$ , of a continuous random variable  $X$  is defined by  $\Pr\{x \leq p\} = p$ , e.g. 0.5 is called the median of the distribution. We use the term "top  $(1 - p)$  quantile" to denote the area  $[p; 1)$ , e.g. top  $(1-0.9)$  quantile means the largest 10% of data. As a running example, let us assume there are 5 peers  $\{P_1; P_2; P_3; P_4; P_5\}$  in the network, and the similarity measures between  $P_1$  and all other peers are  $\{1; 3; 2; 4\}$ , respectively, where the higher the value, the higher the similarity. If  $P_1$  knows the similarity between him and  $P_5$  is 4, our approach will enable  $P_1$  to know, with high confidence, that  $P_5$  is among the top 25% most similar peers of  $P_1$ 's in the network, without computing all the similarity values.

Now we formally define a Peer-to-Peer community based

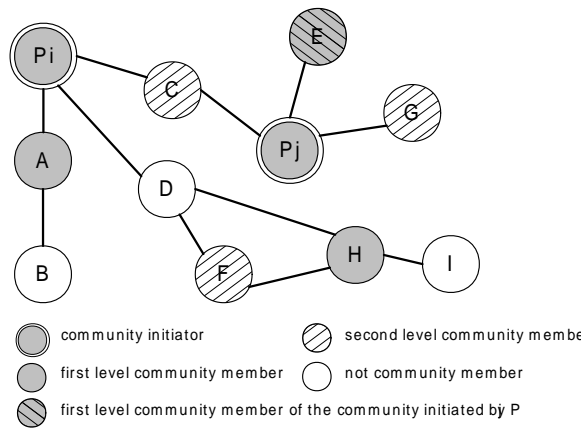


Figure 2: Example of Peer-to-Peer communities.

on our above discussion.

**Definition 3.1.** [( ; p; q)-P2P Community] A ( ; p; q)-P2P community from peer  $P_i$ 's view is a collection of peers in the network, denoted by  $C$ , such that the similarity measures between  $P_i$  and all the members in  $C$  are among the top  $(1 - p)$  quantile of the population of similarity measures between  $P_i$  and all the peers in the network, with confidence level  $q$ .

**Definition 3.2.** [Extended ( ; p; q)-P2P Community] An extended ( ; p; q)-P2P community from peer  $P_i$ 's view is the union of  $C$  (defined by Definition 3.1) and all the peers from the ( ; p; q)-P2P community of each member in  $C$ .

These two definitions implicitly capture the hierarchical characteristics of the community. When a peer finds a similar buddy, he could compute the quantile value and determine which area this buddy belongs to. A peer could also specify a  $p$  value and only invite those belonging to top  $(1 - p)$  quantile area to be his community members. The community could be expanded to include members from members's community. For example, in Figure 2, Peer A;  $P_j$ ; H are the first level members (with larger  $p$ ) of community initiated by  $P_i$ . Peer C; F and G are the second level members (with smaller  $p$ ) of community. Note that  $P_j$  is also a initiator of another community, and it has E as its first level community member. Peer A;  $P_j$ ; H; E compose an extended P2P community initiated by  $P_i$ .

In this paper, we use the scalar product between two profile vectors to quantify the similarity between two peers. Other similarity metrics such as Euclidean distance can also be applied in our framework without any hurdle. Details on how to determine the threshold for quantiles using order statistics theory are given in Section 5.1.

### 3.3 Privacy Management

A major drawback of most existing community formation approaches is that none of them take serious consideration to protect a peer's privacy. For example, a peer may not want to reveal some of his interests, or the weights of his interests in his profile vector. Privacy becomes an extremely important issue especially when the profile is implicitly discovered from the peer's personal activities.

In our framework, we provide the peer with two-level privacy protection. The first level allows the peer to explicitly

filter out extremely private sensitive interests by assigning zero weights to the corresponding concepts in the profile vector. The second level protection relies on the notion of cryptographic secure multi-party computation (SMC) [31]. Loosely speaking, SMC considers the problem of evaluating a function of the private inputs from two or more parties, such that no party learns anything beyond what can be implied from the party's own input and the designated output of the function. We adopt private protocols that are proved to be cryptographic secure such that any pair of peers can compute the similarity of their interests without knowing each other's actual profile. Details about the private computation are given in Section 5.3.

We need to note that no protocols can build a similar interest-based community without revealing the information that these peers share the same interests. A high similarity value between two peers tells them they have a lot in common, and their profile vectors are close. Nevertheless, SMC-based protocols can guarantee that neither party would know the other's actual input, namely, the actual profile vector. Moreover, if the similarity value is low, no significant information about the other peer's interest is disclosed.

## 4. COMMUNITY FORMATION PROCESS

In this section, we address the Peer-to-Peer community formation process under the assumptions that: 1) each peer can be a member of multiple virtual communities; 2) peers interact with each other by submitting or replying queries to determine the potential members of a given community; and 3) there is no superpeer as a centralized authority.

The Peer-to-Peer community emerges as a peer  $P_i$ , called community initiator, invokes a community discovery process which consists of the following tasks: sample size computation, quantiles estimation, member identification, member notification and acceptance, and community expansion.

**Sample Size Computation:** The initiator  $P_i$  first selects a confidence level  $q$  and the order of population quantile  $p$  it would tolerate. Based on the algorithm described in Section 5.1, the initiator calculates the number of samples required to compute the threshold such that any peers that have similarity values with  $P_i$  greater than this threshold are among the top  $(1 - p)$  quantile most similar peers of  $P_i$ 's. Let us denote the sample size as  $N$ .

**Quantiles Estimation:** Given the sample size  $N$ , the initiator invokes  $N$  random walks using the protocols described in Section 5.2 to choose independent sample peers in the network. Whenever a new peer  $P_j$  is chosen, it replies to  $P_i$  with its address and port number, and builds an end-to-end connection with  $P_i$ . Then  $P_i$  computes the scalar product of its profile vector and  $P_j$ 's profile vector using the private scalar product protocol described in Section 5.3. This private protocol guarantees that neither  $P_i$  nor  $P_j$  could know the other party's profile. After  $P_i$  collects all the  $N$  scalar products, it finds the largest one as the threshold for quantiles of order  $p$ .

**Member Identification:** The initiator  $P_i$  composes a discovery message containing its address and port number, as well as a time-to-live (TTL) parameter defining the maximum number of hops allowed for

the discovery propagation. Then the discovery message is sent to all  $P_i$  neighbors. When a peer  $P_j$  receives this message, it replies to  $P_i$  with its address and port number.  $P_i$  then invokes a private scalar product computation (to be described in Section 5.3) to get the similarity value. Independently from the similarity computation and if  $TTL = 0$ ,  $P_j$  forwards the discovery message to all its neighbors, except for the peer from which the message has been received. Each peer discards duplicate copies of the same discovery message possibly received.

**Member Invitation and Acceptance:** The initiator  $P_i$  evaluates the quality of the discovered peers by comparing the similarity values with its threshold. If the similarity is above the threshold,  $P_i$  sends an invitation message to that peer. If the similarity is below the threshold,  $P_i$  still could analyze, with the same confidence level, the order of quantile that the peer belongs to; but note that this order will be lower than the preset  $p$ . Given this information,  $P_i$  can decide whether to send an invitation to a peer with less similarity. For the sake of simplicity, in our experiments,  $P_i$  will not send invitations in this circumstance. Once a peer  $P_j$  receives an invitation message, it decides whether to accept it or not by replying an acceptance message. Receiving the acceptance message,  $P_i$  records  $P_j$  in its local cache.

**Community Expansion:** When a peer  $P_j$  accepts the invitation, it replies to the initiator an acceptance message, as well as the member lists in its local cache. These members are from the P2P community or extended P2P community initiated by  $P_j$ . As a reward, the initiator sends the current member list in its local cache to  $P_j$ . In this way, each peer has an extended Peer-to-Peer community.

In our framework, peers interact with each other by sending discovery queries, or by answering queries. If a peer  $P_i$  polls another peer  $P_j$  but does not get a reply in a reasonable amount of time,  $P_i$  simply assumes that  $P_j$  has left the network. In this case,  $P_i$  can resend query to other peers to get necessary information.

## 5. BUILDING BLOCKS

This section elaborates on some building blocks that are necessary to complete the Peer-to-Peer community formation process.

### 5.1 Distribution-Free Confidence Interval for Quantiles

Given  $x$ , a feature vector, and  $Y$ , a set of other feature vectors, we want to find out how similar  $x$  and a  $y \in Y$  are to each other in comparison with the similarities of  $x$  and other  $y$ s in  $Y$ . A trivial approach to this problem would be to collect the entire set of  $Y$  and compare all the scalar products of  $x$  and  $Y$ . This simple approach, however, does not work in a large-scale distributed P2P environment because the network state is not stable with frequent nodes arrivals and departures, and the overhead of communication would be extremely high. Theories from order statistics, however, could relieve us from this burden by considering only a small

set of samples from  $Y$  and producing a solution with probabilistic performance guarantees. The following part of this section discusses this possibility.

Let  $X$  be a continuous random variable with a strictly increasing cumulative density function (CDF)  $F_X(x)$ . Let  $x_p$  be the population quantile of order  $p$ , i.e.  $F_X(x_p) = p$ . Suppose we take  $N$  independent samples from the given population  $X$  and write the ordered samples as  $x_1 < x_2 < \dots < x_N$ . We are interested in computing the value of  $N$  that guarantees

$$Pr\{x_N > x_p\} > q;$$

Since

$$\begin{aligned} Pr\{x_N > x_p\} &= 1 - Pr\{x_N \leq x_p\} \\ &= 1 - Pr\{\text{all the } N \text{ samples} \leq x_p\} \\ &= 1 - p^N; \end{aligned}$$

we have

$$1 - p^N > q \implies N \geq \frac{\log(1 - q)}{\log(p)} \quad (1)$$

For example, for  $q = 0.95$  and  $p = 0.80$ , the value of  $N$  obtained from the above expression is 14. That is, if we took 14 independent samples from any distribution, we can be 95% confident that 80% of the population would be below the largest order statistic  $x_{14}$ . In other words, any sample with value greater or equal to  $x_{14}$  would be in the top 20 quantile of the population with 95% confidence. The smaller the  $p$  is, the smaller the  $N$ , e.g. when  $p = 0.70$ ,  $N = 9$ . Therefore, given 14 samples, we can also determine the threshold for any quantile of order less than 0.80. Recall in the community formation process, if the initiator  $P_i$  finds a peer  $P_j$  with similarity value less than the threshold, the initiator cannot say  $P_j$  is among the top  $1 - p$  quantile most similar peers, but the initiator can still find out a smaller  $p^0 < p$  and determine with the same confidence level that  $P_j$  is among the top  $1 - p^0$  quantile most similar peers. For detailed treatment of order statistics, we refer the reader to David's book [6].

When  $X$  is discrete, the equation  $F_X(x) = p$  does not have a unique solution. However,  $x_p$  can still be defined by  $Pr\{x < x_p\} < p < Pr\{x \leq x_p\}$ . This gives  $x_p$  uniquely unless  $F_X(x_p)$  equals  $p$ , in which case  $x_p$  again lies in an interval. It can be shown that in this case,  $Pr\{x_N < x_p\} = I_p(N; 1) = p^N$ , where  $I_p(N; 1)$  is the incomplete beta function. Therefore, in the discrete scenario, we have

$$\begin{aligned} Pr\{x_N < x_p\} &= 1 - Pr\{x_N \geq x_p\} \\ &= 1 - p^N > q; \end{aligned}$$

This does not change the conclusion from Eq. 1.

### 5.2 Random Sampling

Random sampling in the networks is a prerequisite to the estimation of population quantile. It can be performed by modeling the network as an undirected graph with transition probability on each edge, and defining a corresponding Markov chain. Random walks of prescribed length on this graph produce a stationary state probability vector and the corresponding random sample. The simplest random walk algorithm chooses an outgoing edge at every node with equal probability, e.g. if a node has degree  $v$ , each of the edges is traversed with a probability  $0.2$ . However, it can be shown that this approach does not yield a uniform sample of the









